

## CSE 231 Project Description

This is Advanced Programming project. Here you will have a chance to get your hands on OOP using Java programming language. You will work with files (excel files) and GUIs. You will implement a Java program that will help Fathy, a fresh graduate software engineer, to finish his tasks in a faster and more convenient way than he does now.

### Overview

Fathy works at great tech company that he always dreamt of working for. He is namely a web developer. He deals with requests sent to web servers and responses from them. You can find detailed explanation for what Fathy does in the prerequisites section. He likes his job. Actually, not that much. He was surprised when he found that he must do some boring systematic work before he can really start writing code. For the systematic work, Fathy is given an excel spreadsheet which represents the web request and response that some specific application may use. The spreadsheet form of representing requests and responses is not a machine-readable one, so, he cannot start developing the web application using the framework that his company uses right away. His boring task is to transform this excel sheet to another form that can be parsed by the framework to produce web API (find more in prerequisites section). **Fathy wants to write a Java program that reads the excel file and identifies main objects in the request and response of a given proposed API(s). You are asked to write this program for him because he is busy with other boring systematic work.**

### Main Requirements

1. Write a utility class that has method(s) can read from excel spreadsheet and probably return an object representing the current excel sheet. (2 Marks)
2. Write classes that represent the main real objects in the problem. You may assume that they are **Operation** (API), **Service** (contains collection of operations), **Field**. This design choice is not a must, but you must justify any other design choice you make other than this one. (3 Marks)
3. Try to output the JSON objects in both request and response of given API(s). You have to use JavaFX or any other Java GUI framework you find convenient. You can log the JSON objects using the standard output, but your marks will be decreased. (2 Marks)

- Write a document that shows your work. You have to show input/output samples of your program in action in addition to the UML class diagram that shows the structure of your classes. (3 Marks)

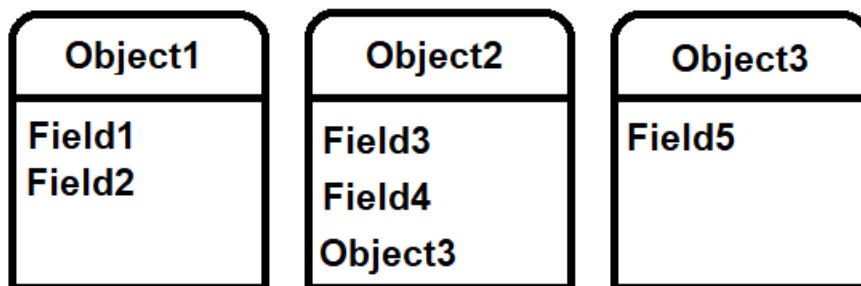
### Important Details and Hints

- If you assume anything rather than the details specified in this document, you have to justify it.
- The work may be done in a team of 2 ~ 4 individuals.
- This is how you can expect the sheet

	A	B	C	D	E
1	REST Operation Mapping (API_NAME)				
2	HTTP Operation	REST URL			
3	PUT	/somePath/somePath2			
4					
5					
6	I/o	Field Name	Type	Allowed Values	Mandatory
7	I	/object1	object1		Y
8	I	/object1/field1	string		Y
9	I	/object1/field2	string		Y
10	I	/object1/field3	string		Y
11	I	/object1/field4	string		N
12	I	/object1/object2	object2		Y
13	I	/object1/object2/field5	string	1,2,3	N
14	I	/object1/object2/field6	string	Y,N	N
15	I	/object1/object2/field7	string		N
16	I	/object1/object2/field8	string		N
17	I	/object1/object2/field9	string		N
18	I	/object1/object2/field10	string		Y
19	I	/object1/object2/field11	string	0,1	Y
20	I	/field12	string		Y
21	O	/object3	object3		Y
22	O	/object3/field12	string		Y
23	O	/object3/object4	object4		Y
24	O	/object3/object4/field13	string		Y
25	O	/object3/object4/field14	string		Y
26	O	/object5	object5		N
27	O	/object5/field15	string		Y

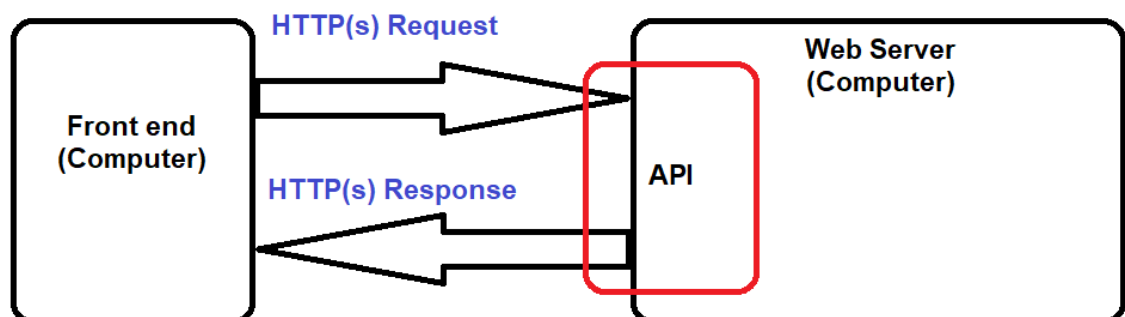
- Objects like object1 is normal fields but can have children like field1.
- Every API has a name that always can be found by iterating over sheet rows and checking if the first cell in the row (cell 0) contains the API name. You have to save the index of the row where you found the API name as it may make it easy for you when you are getting the name of the fields.
- HTTP operation and REST URL are not required attributes of the class Operation (API), but you may make them attributes and store the values given in the sheet.
- I/O column is where you can discriminate between request and response objects (fields) (I stands for Input, i.e., request and O stands for Output, i.e., response).
- The main requirement of this project is that you save all objects (object1, object2, etc.) that belongs to one API.**

- 3.6. The “type” column helps you distinguish between fields that has children (objects) and fields that do not (normal string).
- 3.7. The “allowed values” column is used for strings (fields without children) only. The cells are usually empty which corresponds to allowing “all” values to the string. When a cell contains some comma separated values, this means that the corresponding string is only allowed to have these values (you have to store these values in the field class representation).
- 3.8. The “mandatory” column tells us whether the corresponding field is required in the request or the response. You also have to store this.
4. Hint: For reading excel, you can use Apache POI Java library. Here you can find it in the maven central repository (use the XML as it is explained in the tutorials): <https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml/5.2.2>
5. Here is an expected output sample (does not correspond to the sheet snapshot above and does not have to follow the same ugly UI design)



## Prerequisites

1. Basic web concepts:



- 1.1. HTTP (Hyper Text Transfer Protocol) is the most used protocol to transfer data across the internet. HTTP Verb or HTTP operation is one

of some predefined options that defines the operation the client wants to perform. You can read more about HTTP protocol here (though it is not required to understand or complete the project):

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>

- 1.2. API (Application Programming Interface) is the intermediary between two general computer programs. In our case, the interface is between a web server and a front-end computer. The excel spreadsheet that Fathy got to work with is called the API specifications which allows him to call the back end the way it understands. The spreadsheet clarifies the fields that may be sent in the request (denoted by I) and the expected back-end response (denoted by O).
- 1.3. Java Script Object Notation (JSON) is a format to interchange data while keeping readability for both humans and machines. You can know more about JSON from: <https://www.json.org/>